The opportunistic use of a plan in a design activity : an empirical study of specification

Willemien Visser

UNITE DE RECHERCHE
INRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.(1) 39 63 55 11

Rapports de Recherche

N° 1035

Programme 8

THE OPPORTUNISTIC USE OF A PLAN IN A DESIGN ACTIVITY : AN EMPIRICAL STUDY OF SPECIFICATION

Willemien VISSER

Mai 1989

RR-1035

THE OPPORTUNISTIC USE OF A PLAN IN A DESIGN ACTIVITY: AN EMPIRICAL STUDY OF SPECIFICATION

L'UTILISATION OPPORTUNISTE D'UN PLAN DANS UNE ACTIVITÉ DE CONCEPTION: UNE ÉTUDE EMPIRIQUE DE LA SPÉCIFICATION

Programme 8

Willemien VISSER

April 1989

THE OPPORTUNISTIC USE OF A PLAN IN A DESIGN ACTIVITY: AN EMPIRICAL STUDY OF SPECIFICATION*

Willemien VISSER

* The author wishes to thank André Bisseret and Pierre Falzon for helpful comments on earlier versions of this paper.

Abstract. An observational study was conducted on a mechanical engineer throughout his task of defining the functional specifications for an automatic machine tool installation. The engineer described his activity as following a hierarchically structured plan. This plan, however, does not represent the real activity which is, in fact, opportunistically organized. The engineer follows his plan as long as it is interesting from a viewpoint of cognitive cost. As soon as other actions become more interesting, he abandons his plan to proceed to these actions. This paper analyzes when and how these alternative-to-the plan actions arise. Quantitative results are presented with regard to the degree of plan-deviation, the design components and the definitional aspects which are most concerned by this deviation and the deviation patterns. Qualitative results concern the processes leading to alternative-to-the plan action-propositions which, if selected, lead to plan-deviation. Implications of these results for assistance tools are discussed briefly.

Keywords. Design activity, Specification, Planning, Opportunistic organization of activity, Control, Real-time observational study, Field study, Protocol analysis, Automatic machine tool installation.

Résumé. Des observations ont été conduites sur un mécanicien tout au long de sa construction des spécifications fonctionnelles pour une installation automatisée de machine-outil. Quand on demande au mécanicien de décrire son activité, il la présente comme suivant un plan structuré hiérarchiquement. Ce plan ne rend cependant pas compte de l'activité réelle telle qu'elle a été observée. L'activité réelle est organisée de façon opportuniste. Le mécanicien ne suit le plan qu'il a énoncé que tant que plan est intéressant d'un point de vue de coût cognitif. Dès que d'autres actions que celles dictées par le plan sont plus intéressantes de ce point de vue, le mécanicien abandonne son plan pour procéder à ces actions. Cet article analyse quand et comment surviennent ces actions différentes de celles imposées par le plan. Les résultats quantitatifs qui sont présentés concernent le degré de déviation du plan, les composants de conception couverts par ces déviations et leurs différentes configurations. Les résultats qualitatifs concernent les processus qui conduisent à des déviations. Une brève discussion des résultats à la lumière d'éventuels outils d'assistance clôt l'article.

Mots-clefs. Activité de conception, Spécification, Planification, Organisation opportuniste de l'activité, Contrôle, Observation en temps réel, Etude de terrain, Analyse de protocoles, Installation automatisée de machine-outil.

SUMMARY

1. INTRODUCTION

This paper presents the most important results obtained in a study on the cognitive aspects of specification. The study has been conducted on a mechanical engineer who had to construct the functional specifications for an automatic machine tool installation. Two parts may be distinguished in the installation: an operative part, doing the work - that is, the tooling of connecting rods - and a control part which makes it an *automatic* machine tool installation. This control part is made up by a programmable controller (that is, a computer specialized in controlling industrial processes).

The specifications produced by the engineer were to define the functioning of the operative part for a programmer to design the program for the control part.

Relevance of the study

The study of a specification activity may be considered relevant for several reasons:

Activity of specification: a new contribution to the study of design. Specification as an activity has not yet been studied. Empirical design studies start after the construction of the specifications. The incompleteness of these specifications - a characteristic of design - may be more or less extreme, but the observed designers receive specifications at the start. In this study, the engineer constructed specifications - that is, the definition of a machine tool's functioning - starting from the client's requirements, and the mechanical specifications of the machine tool.
Specification is, however, clearly a design activity, so one should expect to obtain results confirming, or making more specific, results obtained in other design studies.

The importance of the specification stage in design. For two reasons, at least, the specification stage is very important:
- It is a very early design stage, and the activity carried out during this stage has consequences for all the design activity which is done afterwards (see, for example, Letovsky, Pinto, Lampert, & Soloway, 1987, who show the importance of "Design Reconstruction" during code inspection).
- As the specification stage defines the starting point for the following stages, its result is of crucial importance for the resulting design, and thus, so is the activity which leads to this result.

These points hold even if design does not consist of independent, consecutive stages, as empirical studies of design activity tend to show (contrary to the presentation of design in prescriptive studies) (see Visser, 1988a).

Originality of the study

The original aspects of this study are the following.

Study of a professional designer in his daily work context, during his work on a real, industrial design project. Empirical design studies conducted until now have been mostly carried out in an artificially limited context, generally the psychological laboratory. Even if some studies used professional designers (novices and/or experts), these studies generally concerned design tasks which were simplified and somewhat limited compared to real design projects. Moreover, the subjects in these studies were working individually. In reality, design is characterized by people working together and depending more or less on the work done by colleagues and on the information they provide. So, in order to model real design activity, empirical studies of this activity must also be conducted on designers working on real, and therefore complex, projects. Compared with laboratory studies, this type of study is, without a doubt, much more time-consuming, and the data obtained may be more difficult to process, due to the number of factors that cannot be eliminated during observation, as may be done in an experimentally controlled study.

Real-time, continuous observation during the complete construction of the specifications. The engineer was observed throughout the duration of his work (three weeks), until he had completed the construction of the specifications.

Context of the study: a longitudinal design study. The study on the engineer during his specification activity was the first of three studies which together constituted a longitudinal study. This study was carried out on a project to design a computer-controlled machine tool installation (see above).
Various stages in this design process could have been observed. As the study presented here was carried out in the framework of research on programming, the stages observed were those that were considered the most relevant from this point of view. Besides the stage which concerned programming as such (presented in Visser, 1987), the stages in the design process that immediately preceded and followed it were also observed. The three studies were thus conducted on:

- the construction of the functional specifications of the operative part of the installation (the study presented in this paper). This work was done by a mechanical engineer, and the functional schema that he produced constitutes the main document used to provide the specifications for designing the program to control the installation;
- the writing of this program by a specialist in electronics and automation (the "programmer");
- the debugging and testing of this program by another specialist in electronics and automation.

Focus of this paper: opportunistic plan-deviation actions. The study which is presented in this paper focuses on the way the engineer organizes his specification activity. The engineer describes his activity in the form of a hierarchically structured plan. He breaks his activity down into components of different levels to be dealt with in a specific order. This plan, however, does not represent the real activity which is, in fact, opportunistically organized. The engineer follows his plan as long as it is interesting from a viewpoint of cognitive cost. As soon as other actions are more interesting on this point, he abandons his plan to proceed to these actions. This paper analyzes when and how these alternative-to-the plan actions come up.

* * *

Organization of the paper. Section 2 describes the method used to collect and analyze data on the specification activity. Section 3 presents the main results, distinguishing quantitative results on the degree of plan deviation observed and qualitative results concerning the processes leading to plan-deviation actions. Section 4 concludes with a short discussion of the implications of these results for design assistance tools.

2. METHOD

2.1 AN OBSERVATIONAL STUDY

During a period of three weeks, full time observations were conducted, in a machine tool factory, on a mechanical engineer involved in a specification task. The engineer's normal daily activities were observed without any interventions, other than to ask him to verbalize as much as possible his thoughts about what he was doing (see Ericsson & Simon, 1984; Newell & Simon, 1972).

2.2 THE OBSERVED TASK

The engineer which has been observed had to specify the functioning of the operative part of an automatic machine tool installation described below. The functional schema he produced was intended to specify this functioning for the programmer of the control part of the installation. The specifications were to be produced in the form of a particular type of functional schema, a Grafcet (described below).

2.2.1 Physical layout of the operative part

The operative part of the installation is composed of four "stations." A central turntable rotates to bring and position the pieces in front of each of them. The pieces are conveyed onto and from the loading-unloading station by a system of conveyors (see Figure 1).

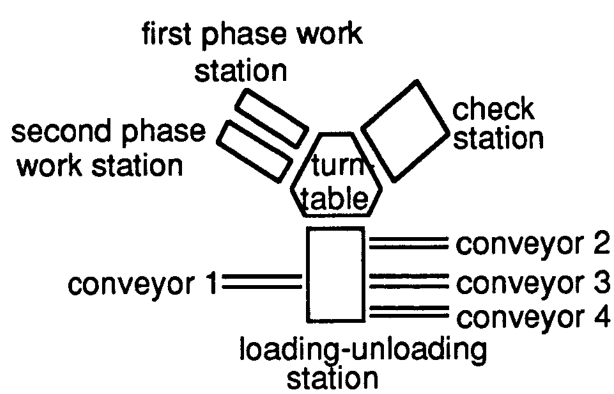

Figure 1. Spatial organization of the operative part of the machine tool installation

For mechanical reasons, the Second phase workstation precedes the First phase workstation in the rotational direction of the turntable. The connecting rods which are going to be made on this installation are however first shaped on the First phase workstation, and then finished on the Second phase workstation. It will be seen that this arrangement has consequences for the specification activity.

2.2.2 The engineer's informational starting point for specifying: functional requirements, mechanical specifications and an example of specifications

For constructing his specifications, the engineer starts with two types of information about the operative part of the installation: Functional requirements and Mechanical specifications. He also uses the specifications document of an analogous machine tool installation (in a sequences schema formalism, see below).

Functional requirements. These are mainly the client's requirements, but also a first, global analysis of them which has been made by mechanical design colleagues, in the so-called "Pre-Study" stage. These designers especially listed the operations which they judged to be needed for the required functioning. This analysis is reflected in one of the mechanical specifications documents (see below), the two workstations "Tool-plates": next to a rough drawing of each station, these plates present the operations, along with their duration and the technical specifications of the motors and of the tools which control them (their rotation speed, their power, and their advance distance/rotation).
These tool-plates will be consulted, modified and completed by the engineer.

Mechanical specifications. They specify
- the mechanics of the machine tool (its stations, their tools, the motors which govern their forward/backward movements, their physical devices, such as the jacks and detectors);
- the material possibilities of these mechanics: stations and their tools may go forward and backward, at a faster or slower rate, etc.

Specifications of an analogous machine tool installation (the "example"). Before he starts his specification activity, the engineer looks for an example of specifications which are analogous to the specifications which he has to construct. So, he takes the sequences schema of an analogous machine tool installation which has been previously constructed by a colleague.

The engineer's use of these documents. The first two types of documents specify the mechanical *possibilities* of the operative part - and some design hypotheses, in the form of possibly required operations. The engineer, specifying the functioning of the operative part, has to make *choices and decisions* as to which of the various possibilities will be implemented and how this will be done.

For example, the connecting rods which have to be made on the machine tool installation have two extremities. Both have to be tooled, but they do not undergo the same operations. Specific tools are present on each workstation with separate control organs. So, the engineer has to decide which operations are going to take place on the different ends of each piece, and the articulation between these operations (for example, are they going to take place simultaneously or successively?).

The example is sometimes used as a reference when the engineer has a problem. Then he consults this example for examining the solution it proposes.

2.2.3 Specification formalism: a functional schema

Several representation formalisms are used in industry for specifying sequential processes. In the industrial plant where the observed engineer is working, the formalism that had always been used before this study was a "sequences schema". The installation the specification of which has been studied was the first to be specified in the "Grafcet" formalism, because it is supposed to allow for less ambiguity (see below).
The observed engineer decides however to proceed in two stages:
• to specify the installation functioning in his usual formalism, that is, a "sequences schema";
• to "translate" this schema into a Grafcet.
Both stages have been observed. This paper presents only the first stage, except for some global comparisons made with the second one. The Grafcet construction has been presented elsewhere (see Morais, 1987).

As the engineer, for his specification activity, uses a sequences schema, this representation formalism will be described with the detail required for understanding the results. For the Grafcet formalism, its most important characteristics will be presented.

> NB. It was the engineer's intention to do the specification on a sequences schema, and to "translate" the result into a Grafcet. During this "translation" stage however, he still took design decisions not yet taken (and made numerous modifications to the specification as it was reflected by the sequences schema). He defined, for example, a function for which the mechanical requirements had not yet been set by the client when the engineer was constructing the sequences schema (the conveyors function).

Sequences schema

This formalism represents an installation functioning according to a decomposition at three levels: the Installation - the installation Functions - the function Operations. Materially, it is represented on "Cycle-plates". These plates are called "Cycles" by the designers, but "cycle" also refers to the functioning represented on a plate. In this text, "cycle" will be used to refer to the functioning, "cycle-plate" to its representation.

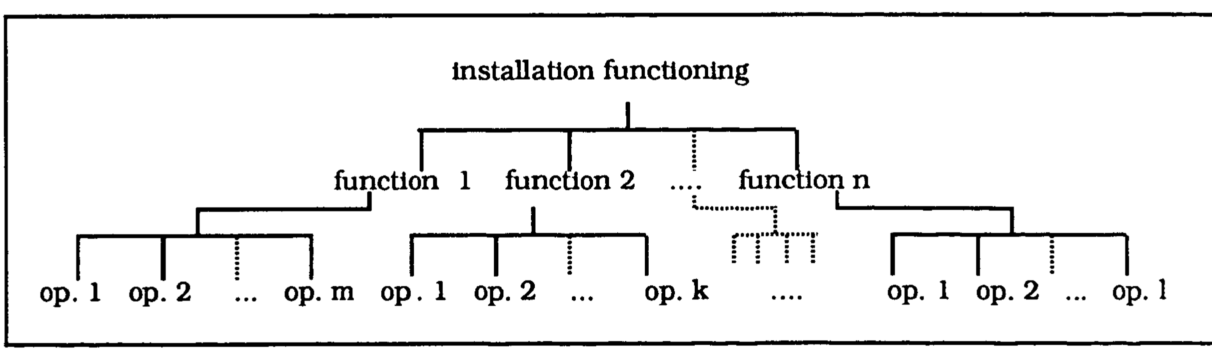

Figure 2. Decomposition of the installation functioning

Design components. The installation functioning is decomposed into what will be called in this text "components"[1]. At the highest level, it is decomposed into functions. Each function is realized by a sequence of operations. The installation functioning is cyclic, at the level of the functions, and therefore also at the level of the function operations (see Figure 2, p. 5).

There are two types of cycles, and corresponding cycle-plates. There is one General Cycle-plate representing the different functions of the installation, along with their defining descriptors and their temporal articulation. For each function, there is a Function Cycle-plate representing the operations fulfilling this function, along with their defining descriptors and their articulation (see Figure 3).

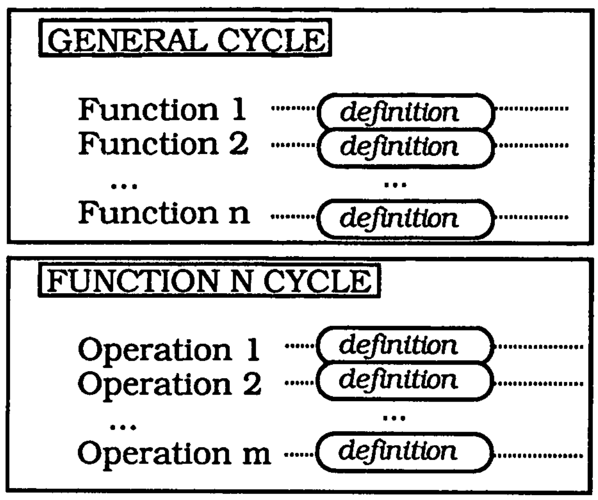

Figure 3. Schematic representation of the two types of cycle-plates: the General Cycle-plate and the Function Cycle-plates

Functions, as well as their operations, are defined each by way of several "descriptors" (for example, their duration, or their starting and ending conditions). The temporal articulation between the functions (in the General Cycle) and the operations (in each Function cycle) is represented graphically. This will be illustrated in the example of a Function Cycle-plate.

Example of a Function Cycle-plate. Figure 4 (see p. 7) presents the example of the Turntable Cycle-plate. It is a partial representation, only presenting the elements on a Function Cycle-plate which are relevant for explaining how an operation is defined. It shows especially the representation formalism used for the different dimensions of an operation's definition:

- Identifier. The "identifier" of an operation is written in the column "Operation";
- Duration. Its "duration" is represented in two ways: (a) in the form of a numeral in the "Time"-column, written before (b) its analogical representation, a line of a length which is proportional to this duration. The duration of the operation Fast Movement, for example, is 3,5 time units (1/100 minute), represented by a line of 7 time-representation units.
- Articulation between operations. This articulation is represented through the relations between the lines representing the different operation durations. For two operations which take place simultaneously, their duration lines will

1 "Components" is a term introduced by us, to refer - without distinction - to "Cycles", "Functions", and "Operations" - terms used by the engineer.

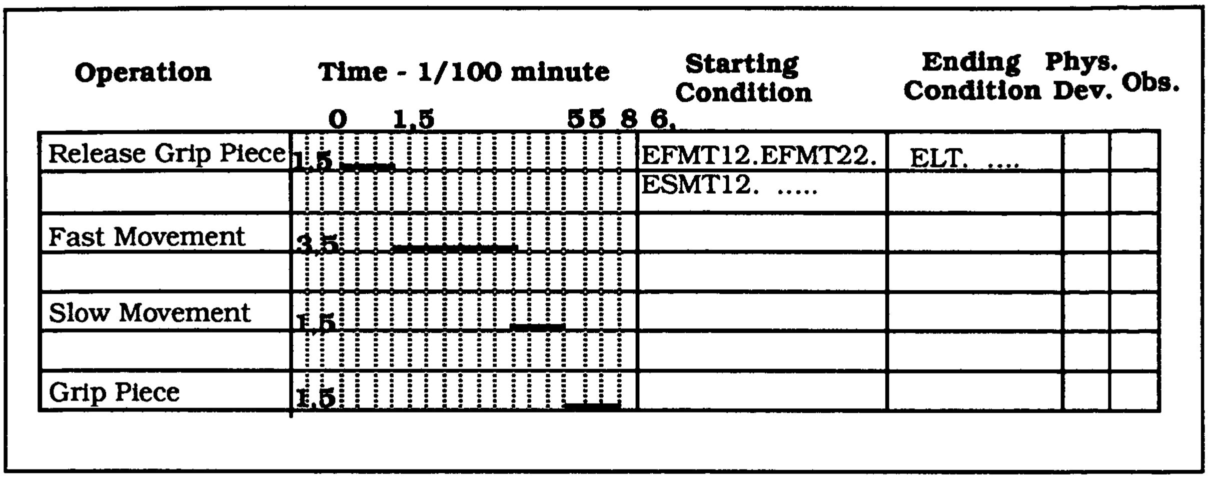

Figure 4. Representation of the Turntable Cycle-plate (part)

be parallel; the duration line of an operation which follows another one will start there where the duration line of the other operation ended. So, the operation Fast Movement, for example, follows the operation Release Grip Piece.

- Starting - and Ending conditions. These are represented by boolean expressions: their elements are the mnemonics - that is, the names, in an abbreviated form - of the control detectors which must be activated, or not activated for the operation to start, or to end. The starting conditions of Release Grip Piece, for example, are: End of Fast Movement of Table12 (EFMT12) AND End of Fast Movement of Table22 (EFMT22) AND End of Slow Movement of Table12 (ESMT12) etc.
- Physical device. In this column figures the mnemonic corresponding to the motor or jack for activating or inhibiting the operation.
- Observations. They may be present or not for an operation. The engineer used this column for representing several things: ending or starting conditions for which he did not know the materialization; safety conditions which are generally different from the starting conditions of one particular operation, because they apply to all the operations of a function[2]; conditions under which a machine operator has to act on the installation, mostly to stop it, for adjusting or fixing something[3].

Grafcet

This formalism gives a graphical representation of the sequential progress of the process, showing the alternation of its actions and their enabling conditions (corresponding partially[4] to the starting and ending conditions used on the sequences schema). For each function, the graphical illustration is accompanied by documents providing a written description of (a) the safety conditions for each action; (b) the physical starting point of the cycle; (c) the procedure to be

2 The sequences schema-formalism poses a representation problem on this point. It does not provide the possibility to represent conditions applying to more than one operation - other than by repeating these conditions for all operations which are concerned. The Grafcet-formalism allows the representation of this kind of information.

3 On this point, the only possibility is putting the information elements in the Observations-column. Once again, the Grafcet-formalism allows this kind of information to be represented in a more appropriate manner.

4 The "ending conditions" of an operation are not simply the "starting conditions" of the following operation, but generally some of the "ending conditions" of an operation are among the "starting conditions" of the following operation.

action; (b) the physical starting point of the cycle; (c) the procedure to be followed in the event of a mechanical or process control problem (see Morais, 1985).

2.3 THE OBSERVED MECHANICAL ENGINEER

The observed engineer had a professional experience of more than ten years in the machine tool factory. The mechanical installation studied - type tooling - was not his specialty, which was assembling. Tooling and assembling are two applications in machine-tool manufacturing. Most mechanical engineers know both, but are specialized in one of the two. It will be seen that certain results may be attributed to this characteristic of the subject.

2.4 DATA COLLECTION

Notes were taken on the engineer's actions; all documents which he produced during his work were collected.

Notes. These concerned:
- the engineer's actions and the remarks and comments he made;
- the order in which he produced the different documents, and how he gradually built them up;
- the changes he made;
- the information sources he consulted;
- the events considered by the observer to be indicators of the subject meeting with difficulties.

Documents collected. These were:
- the different versions of the sequences schema;
- the diagrams and schemas the engineer constructed for himself during his problem solving.

2.5 DATA ANALYSIS

Among the actions observed on the engineer, only those which contribute to the specification of the installation's functioning are analyzed in the study presented in this paper.

Specification-action unities. Specifying is considered to be a problem-solving activity. The corresponding reasoning chain has been decomposed into problem-resolution steps: each unity corresponds to a specification-action concerning a design component on one dimension.

Example. Defining a function with regard to its identifier and its duration constitutes two action unities.
Verifying the ending conditions of an operation is considered to be one action unity.

Component-processings. A sequence of action unities which concern the same component is considered to be one component-processing: according to the nature of the component, a *function-processing* or an *operation-processing.*

Example. If defining a function with regard to its identifier and its duration constitutes two action unities, it constitutes one function-processing.
Verifying the ending conditions of an operation is considered to be one action unity and one operation-processing, if immediately afterwards the engineer goes to another component. If he also verifies the duration of this operation - without any interruption between the two action unities - this duration verification belongs to the same operation-processing

Definition-sessions. A sequence of component-processings which concern the same higher-level component is called a component-definition-session:

according to the nature of the component, a *cycle-definition-session* or a *function-definition-session* (see Figure 5).

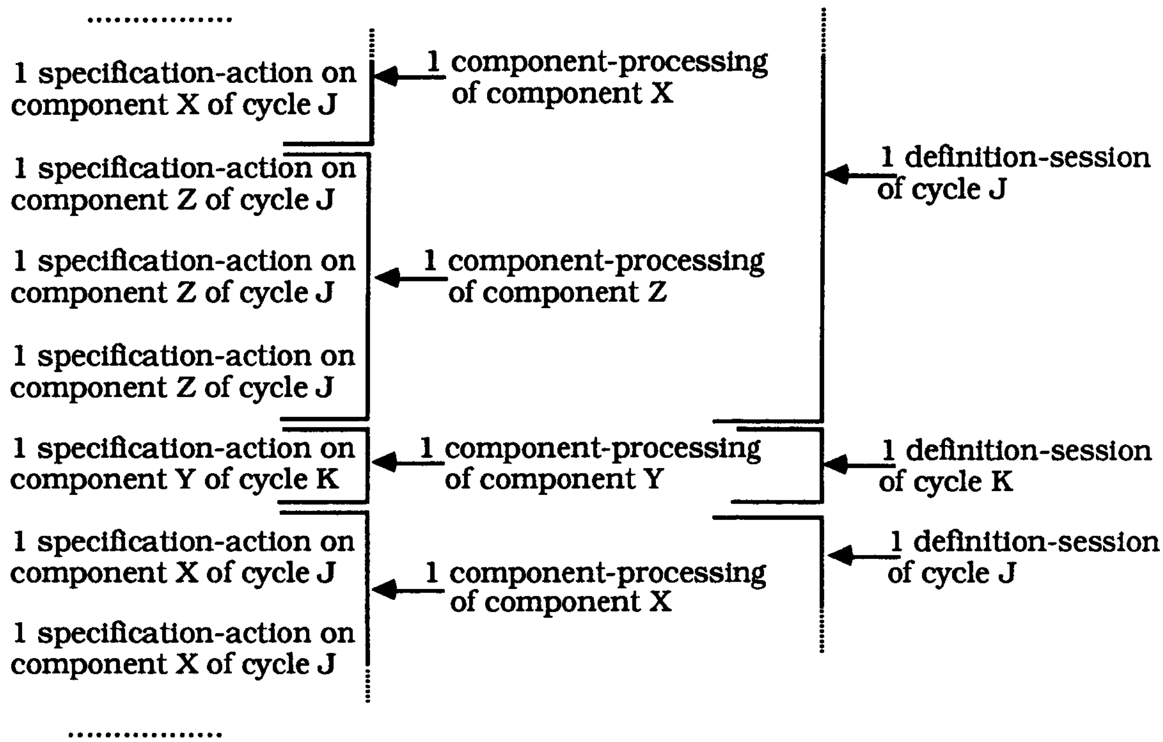

Figure 5. Relation between specification-actions, component-processings and definition-sessions

3. RESULTS

Preliminary methodological remark. Fluent and continuous verbalization. Simultaneous verbalization - as was asked of the observed subject - is not always easy, or even possible, for a person involved in a problem-solving activity. The engineer did not find it difficult, or troubling, to "think out loud". The observer never had to remind him to keep on verbalizing, as is generally required in protocol studies, where subjects may be seen to become lost in thought, or to resume their normal, silent mode of thinking (see Ericsson & Simon, 1984).

> NB. In the observational study conducted on the programmer designing the program from the specifications made by the engineer, the same simultaneous verbalization technique was used (see Visser, 1987). The programmer however verbalized rather little while writing the program. The hypothesis was formulated that many of the programmer's actions during program construction are automatized (as a consequence of his experience in the field) and that verbalizing would require a "decompilation" of these automatized procedures (see Anderson, 1986). Encouraging the programmer to verbalize more might lead him to make the knowledge sources underlying these procedures explicit, but such verbalization would not express the real activity the programmer performs in writing his program.

The engineer's continuous verbalizing may, with some caution, be interpreted as his specifying being an actual problem-solving activity, with only few automatized components.

3.1 GLOBAL ORGANIZATION OF THE SPECIFICATION ACTIVITY

At the level of the organization of the activity, the most important and interesting result of this study is that
- the engineer describes[5] his activity as following a hierarchically structured plan[6];

but
- his real activity, that is the activity such as it has been observed, is opportunistically organized: the engineer follows his plan only as long as he does not perceive more opportune actions (see Hayes-Roth & Hayes-Roth, 1979).

After a short presentation of this plan, and the roles it may play, the rest of this chapter will concern the real specification activity, and focus on plan-deviation.

3.1.1 The plan of the activity as described by the mechanical engineer

The engineer described his activity in the form of a hierarchically structured plan containing four levels (the functioning to be represented on the sequences schema - its cycles - their components - their descriptors) (see Table 1, p. 11). The engineer's control procedure for covering this tree-structure reflects a top-down, depth-first planning.

---

[5] The engineer was requested to describe his activity before carrying it out and afterwards.

[6] We call the plan "hierarchically structured" and not "hierarchical", to avoid confusion with the "hierarchical planning" as described by Sacerdoti (1974).

Table 1. Representation of the description of his activity given by the engineer

```
+ Construct a sequences' schema
  + Define the General Cycle
    + Define its first function
      + Determine the first function
      + Determine its descriptors
        + Determine its identifier (id)
          Insert its value in the id-column
        + Determine its duration (du)
          Insert its value in its du-column
        + Determine its starting conditions (st)
          Insert their values in its st-column
        + Determine its ending conditions (e)
          Insert their values in its e-column
    + Define its second function
      + Determine the second function
      + Determine its descriptors
          ...
        ...
    + Define its nth function
        ...
  + Define the first Function Cycle
    + Define its first operation
      + Determine the first operation
      + Determine its descriptors
        + Determine its identifier (id)
          Insert its value in its id-column
        + Determine its duration (du)
          Insert its value in its du-column
        + Determine its starting conditions (st)
          Insert their values in its st-column
        + Determine its ending conditions (en)
          Insert their values in its en-column
        + Determine its physical device (ph)
          Insert its value in its ph-column
    + Define its second operation
      + Determine the second operation
      + Determine its descriptors
          ...
        ...
    + Define its mth operation
        ...
  + Define the second Function Cycle
          ...
      ...
  + Define the nth Function Cycle
          ...
```

Key :+ precedes a goal to be realized
(the lines not preceded by + describe the actions carried out in order to realize the immediately preceding goal)

... replaces goals whose breakdown follows the breakdown presented higher on in the schema for a similar goal

For reasons given below, this plan does not represent the engineer's real activity. The same is true for both of the other operators: the plans they produce do not conform to their real activity. In general terms, the operators do not follow a systematic path through the tree representing the plan which they produced as describing their activity (see Visser, 1988a).

A plan: Two functions

A plan may guide one's activity in at least two ways (see Hoc, 1988b).

Declarative plan: Structure of the result of the activity. Presenting the structure which the result - or an intermediary state - of the activity must have, a "declarative plan" allows the states this activity must attain - or go through - to be anticipated.

Procedural plan: Structure of the activity. If it represents both the coordination between actions to be realized, and elements of the control structure, a plan may really guide the activity.

The plan presented by the engineer, that is a hierarchical structure organizing actions, accompanied by a procedure which covers it, reflects such a "procedural plan". The engineer may think that he follows it, but he is observed to deviate from it whenever more opportune actions, or more opportune local plans, are perceived. His plan surely guides him, but only as long as no opportunities arise which are more interesting because they are cognitively more economical: if they should come up, then the plan is immediately abandoned.

3.1.2 The real activity as it was observed

The real activity as it was observed will be presented from two points of view. First, the degree to which the engineer deviates from the plan will be quantified (§3.2). Then, the processes leading to these deviations will be presented in the context of a general blackboard model of the specification activity (§3.3).

3.2 QUANTITATIVE RESULTS: DEGREE OF PLAN DEVIATION

Table 2 (see p. 13) gives the abbreviations used in the presentation of the results.

3.2.1 Highest level: Cycle definitions

Preceding methodological remark. Missing data. The observations made on the definition of the Check Function are incomplete: the engineer started it one evening, after work time. This function will be considered, in this chapter, only where qualitative data are sufficient, that is for questions of definition order.

Cycle-definition interruption

The engineer does not complete the definition of a cycle before he starts another one: only half (52%[7]) of the definition of each cycle is done without interruption (between 31% - for W2 - and 73% - for GC).

[7] All percentages presented below are averages. If spreading is important, the extreme values are given also.

Table 2. Abbreviations used

GC	General Cycle
Tt	Turntable Cycle
Ld	Loading Cycle
W1	First phase Work Cycle
W2	Second phase Work Cycle
Ch	Check Cycle
C	component
D	descriptor
O	operation
id	identifier
du	duration
st	starting conditions
en	ending conditions
ph	physical device

On average, he defines a cycle in five cycle-definition-sessions, that is, after he has started the definition of a cycle, and after he has abandoned it for defining other cycles, he still interrupts four times the definition of these other cycles to come back to it. He mainly does so for completing, not for modifying. He completes mostly the information that specifies the start of the cycle (by way of the descriptor "starting conditions" of its first function or operation[8]).

Except for the Loading Cycle, the definition of each cycle is interrupted with approximately the same frequency: about once every four operation-processings, that is, one cycle-definition-session is made up, on average, of some four operation-processings. The Loading Cycle is interrupted much less than the other cycles: its cycle-definition-sessions are made up, on average, of 11 operation-processings. An explanation for this result - along with other ones on the Loading Cycle - will be presented below.

Only for the General Cycle is the greatest part of its definition made during its first cycle-definition-session (73%). For the other cycles, it is during later sessions that the greatest contribution to their definition is made (the third, fourth or sixth session).

One could ask, for the different cycles,
• when these sessions take place
and
• which proportion of its total definition is made during these sessions.

Figure 6 (see p. 14) presents this information. It represents all consecutive cycle-definition-sessions. For those sessions during which more than 5% of the definition of the corresponding cycle is made, the percentage of total cycle

[8] This is not "correct", but the sequences schema does not at all allow the representation of this information. Once again, the Grafcet-formalism does.

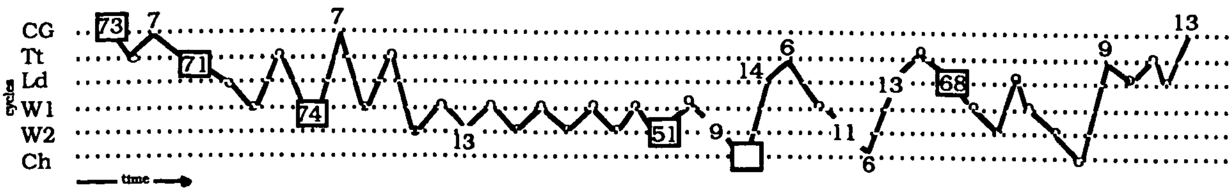

Figure 6. Order of cycle-definition-sessions

Key: ° corresponds to a cycle-definition-session during which ≤5% of its definition is made

x (where x > 5) corresponds to a cycle-definition-session during which >5% of its definition is made, with this percentage

□ corresponds to the cycle's definition-session during which the greatest contribution to its definition is made, with the percentage corresponding to this contribution

NB. The empty case for the Check Cycle's definition-session reflects
• the hypothesis that this greatest contribution was made during the first definition-session of this cycle
• the fact that this session was not observed

definition is mentioned. It shows also, for each cycle, during which session the quantitatively greatest contribution to its definition is made.

Two points were examined for the cycle-definition interruptions:
• transition patterns: what is the nature of the other cycle for which a cycle's definition is interrupted: is there a regularity? and, if so, what type of relation exists between the two of them?
• deviation-patterns: which is the nature of the interruption: is only one other-than-the-plan component processed and does the engineer come back to the interrupted cycle-definition immediately afterwards? or does one deviation bring about another?

Figure 6 shows that only the two Work Cycles have a rather systematic transition-pattern: for each of them, about two-thirds of the interruptions are made for defining the other. The Lowest-level presentation will detail the reasons for these movements.
With regard to deviation-patterns, the Work Cycles are also the only cycles for which an interrupted definition is, in most cases (77%), resumed immediately afterwards.

Function definition order on the General Cycle-plate vs. definition order of the Function Cycles

The order in which the different functions are introduced on the General Cycle-plate is not modified any more afterwards - unlike what occurs for the operations on the function cycle-plates. The engineer plans to define the individual Function Cycles in the same order as he defined them on the General Cycle-plate. In reality, he defines them afterwards in another order (see Table 3).

Table 3. Confrontation between the orders in which functions are defined on the General Cycle-plate and in which Function Cycles are defined

Function definition order on GC	Function Cycle definition order
Tt	Tt
Ld	W1
W2	W2
W1	Ch
Ch	Ld

The observed differences deserve comment on two points:
• the criteria used for ordering the functions
• the moment of definition of the Loading Cycle

<u>Different criteria used for ordering the functions on the General Cycle-plate vs. for ordering the Function Cycles</u>. The functions are introduced on the General Cycle-plate in the order in which the turntable rotates to position the pieces in front of each of the corresponding stations on the installation. The engineer starts at the Loading-Unloading station - which is generally the reference on machine tool installations.

On all requirements and mechanical specification documents and construction drawings, the stations are also numbered according to this rotation order:
• the Loading-Unloading station is numbered 01
• the unity constituted by the two workstations together is numbered 02:
    • the Second phase workstation is numbered 021
    • the First phase workstation is numbered 023
• the Check station is numbered 03
This rotation order however does not correspond to the order in which the pieces are tooled, which is First phase (shaping) before Second phase (finishing).

The tooling order - that is, the functioning order of the installation - is the one followed by the engineer to define the individual Function Cycles. This explains why the Work Functions are defined in the reverse order compared to their definition order on the General Cycle-plate. On the General Cycle-plate they have been defined only globally. For their detailed and precise definition, their functioning is relevant, not the rotation order of the turntable or the numbering of the stations.

<u>Moment of definition of the Loading Cycle: postponement for reasons of cognitive cost</u>. Figure 6 seems to show that the definition of the Loading Cycle is initiated after the Turntable Cycle definition. During this first "cycle-definition"-session of the Loading Function, the engineer however makes no actual definition. He examines the Loading Function on the example sequences schema, does not understand the decomposition it has received[9], and this leads him to decide to postpone the Loading Function definition. So, the engineer defines the Loading Cycle only after having defined the Check Cycle.

[9] In the example, the Function Cycle is decomposed in two parts: "General" and "Detail".

3.2.2 Intermediate level: Global component definitions

Half of the function- and operation-definitions (55%) is finished during its first processing, that is, before the engineer starts defining another component.

For the functions of the General Cycle, an important part of each function is defined during its first processing: on average, 86% of the descriptors used for their definition are attributed. However,
- as will be detailed below, on the General Cycle-plate the engineer defines the functions only on one or two dimensions (id, or id and du);
- only 67% of the descriptors used are set once and for all, the others will still be modified;
- during this first function-processing, 67%[10] of the functions are defined completely, that is, on all dimensions on which the engineer defines them: only one or two (see above).

For the operations of the Function Cycles, 68% of the descriptors used for their definition are attributed during their first processing (between 41% - for Tt - and 81% - for Ld). However,
- only 55% of the descriptors are set once and for all (between 25% - Tt - and 74% - for Ld), the others will still be modified;
- during this first operation-processing, only 57% of the operations are defined completely (between 0% - for Tt - and 67% - for Ch).

Temporal articulation between the operations of a Function Cycle

Using the same criterion as for the definition order of the Function Cycles, the engineer *represents* the components of a Function Cycle - that is, its operations - in their functioning order. Generally, he also *introduces* them in this order, but certain operations are forgotten at a first time, and introduced - and inserted at their appropriate position - only afterwards. This holds especially in the case of the Work Cycles.

- Turntable Cycle. Its four operations are introduced immediately in their functioning order.

- Loading Cycle. On 32 operations introduced "under our observation"[11], three changes have been observed:
  - For two operations, the engineer changes his mind about their temporal articulation with the other operations. For both of them, this judgement intervenes immediately after he has introduced them, that is, during their first processing or during the first processing of its next operation (which he then judges to take place before the one already introduced).
  - A group of two simultaneous operations is forgotten until much later: actually taking place in 13th position, they are only remembered in 30th and 31st position.

- Work Cycles. It is on these two cycles that important omissions and changes regarding the temporal articulation between operations intervene.

---

[10] It is a coincidence that the two percentages are the same (67%):
- the first reflects that 12 among 19 descriptors are not modified any more;
- the second that 6 among 9 functions have received all the descriptors by way of which they are going to be defined (that is, one or two).

[11] The other 20 Loading operations were defined one evening, after work time.

Out of a total of 26 (two times 13), the engineer forgets ten operations, that is, in a first pass.

Two remarks before detailing and commenting these results:

- As mentioned in §2.2.2, for the definition of these cycles, the engineer makes use of two requirements analysis documents, the tool-plates. His most important reference for their definition is however his knowledge of the functioning required and the functional analysis he makes himself based on the client's requirements. The tool-plates are used for retrieving physical information about the operations they present (duration, etc.).
- The Work Cycles operations are of three types:

  Movements of the station (Forward and Backward), during which no work (tooling) is done. The Forward movement is for positioning the station in the tooling departure position. The Backward movement positions the station in a mechanical safety position[12]. There are four of these movements, one Forward movement and one Backward movement on each station.

  Work operations are the operations doing the work. They realize the goal of the process. There are five of them.

Prerequisite operations - that is, prerequisites of the Work operations - are operations preparing the process to attain its goal (for example, positioning a tool, or setting a stop, to establish a position where an operation must end). There are 17 of them.

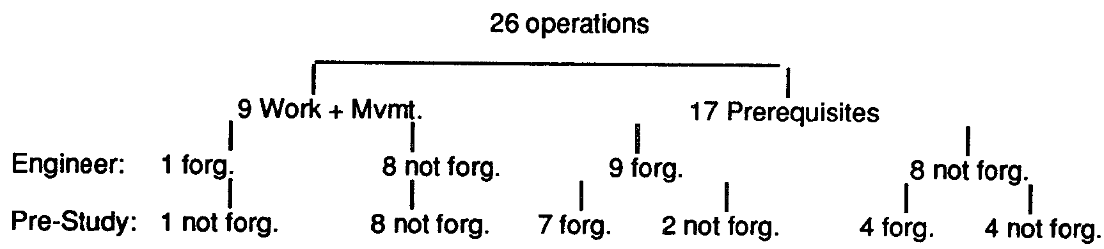

Figure 7. Distribution - among Prerequisites and Work Operations + Movements - of operations forgotten by the engineer and/or by his colleagues during their requirements analysis (the Pre-Study stage)

<u>Operations forgotten are Prerequisites</u>. This holds for nine out of the ten operations which are forgotten by the engineer[13] (see Figure 7). The only other operation which he forgets is a very unusual Work operation: it takes place during the Backward movement of the station. In machine tool installations, work operations, nearly always, take place during the Forward movement of the station.

The engineer often only discovers that he has forgotten a Prerequisite operation when he processes the operation of which it is a prerequisite, an opposite operation, or a similar operation - that is, an operation of which the representation may activate the representation of the forgotten one by way of the relationships the two representations hold in memory (see §3.3.1 below).

NB. The programmer does not forget this peculiar Work operation which takes place during the Backward movement of the station, but he defines it as taking place during the Forward movement - an error which is

[12] In which, for example, the Turntable may rotate without hitting the stations, or tools may be changed.

[13] "Forgotten", that is, forgotten in a first pass, but introduced afterwards.

subsequently corrected by the person doing the debugging. Elsewhere (see Visser, 1987), this result was interpreted as due to *Schema-guided information processing*. The programmer violated the specifications for this "atypical" operation, by defining it as a "prototypical" operation. If he did read the specifications - he was observed to skim only once, and rather rapidly, through them - , his expectations - based on prototypical schema slot values - were probably so strong that he did not take into account the values which were explicitly given (in the specification document). If he wrote the relevant part of the program without reading its specifications, the schema he instantiates may have provided him preferentially with this prototypical value.

Operations forgotten had not been specified before. Out of the nine - prerequisite - operations which the engineer forgets, seven had also not been planned during the Pre-Study (the preceding design stage, see §2.2.2), that is, they do not appear on the tool-plates. Four other operations had not been planned during the Pre-Study, but were not forgotten by the engineer: they also were all Prerequisites (see below, §3.3.1, for a comment on the specific role of prerequisites in information processing).

Modifications during construction of the Grafcet. The engineer's intended strategy was to do the specification in a sequences schema formalism, and then to "only translate" the result into a Grafcet. During this "translation" however, he still made numerous modifications. One important modification type concerned the operations temporal articulation. For the First phase work cycle, he made one operation to take place later in the cycle. The Second phase work cycle was considerably modified on this point: for five - out of 13 - operations, their temporal occurrence was changed, and three of these operations were Prerequisites.

3.2.3 Lowest level: Detailed component definitions

Descriptors used for definition

Components are to be defined by way of five descriptors:
id identifier
du duration
st starting conditions
en ending conditions
ph physical device

Moreover, they may receive observations, in a special "Observations" column (larger than could be represented in Figure 4). This column is sometimes used for putting information which is difficult - or even impossible - to represent on a sequences schema (see Notes 2 and 3).

If all the five descriptors are used, the engineer attributes them - except for 7 of the 83 components - in this order, which is in complete accordance with his plan. However, he generally does not attribute them during the first processing of the corresponding components, nor maintains them without modification, once they have been attributed.

As a matter of fact, even if the specifications are supposed to define all components by way of these five descriptors, the sequences schema produced by the engineer does not specify all components completely. Two remarks may be formulated:

• it may be considered that specifications are always imprecise and incomplete and that this result is "normal";
• it is also possible that the engineer does not consider that the specifications on the sequences schema are complete and that he plans to complete them during their "translation" into the Grafcet formalism.
This is contrary to what he says after finishing the construction of the sequences schema. It could be however that the engineer's verification criteria are less severe than usual for this not-final specification construction, which is going to be followed by a "translation" into another specification formalism.

General Cycle. The functions of the General Cycle are not defined by way of all five descriptors: for half of the functions, two descriptors are used (id and d), for the other half only one (d).

Function Cycles. The operations of the Function Cycles are mostly defined by way of three to five descriptors: 27% is defined by way of only one (id) or two (generally id and d) - 41% by way of five.
With regard to the frequency of attribution of the different descriptors to the operations, the results are the following:
• All operations receive an identifier.
• Duration, starting and ending conditions are attributed in most cases (71%).
• Half of the operations are also defined by way of their physical device (48%).
• Observations are made for 9% of the operations.

Reprocessing an operation: quantitative differences from cycle to cycle

The number of times an operation is processed differs between the functions (see Table 4).

Table 4. Percentage of operations, per function cycle, receiving one or more processings

	number of processings per operation												
	1	2	3	4	5	6	7	8	10	11	tot op	tot pr	av % pr/op
Tt	0	0	20	0	20	20	0	20	0	20	5	33	6,60
W1	44	6	17	11	6	11	0	0	6	0	18	55	3,06
W2	33	28	11	11	6	6	6	0	0	0	18	48	2,67
Ld	48	42	9	0	0	0	0	0	0	0	33	83	2,52
tot	41	27	12	5	4	5	1	1	1	2	74	219	2,96

Key:
tot op	total number of operations
tot pr	total number of processings
av % pr/op	average percentage of processings per operation

Two results in particular will be commented on:
• the decline in the number of processings per operation as definition progresses from the Turntable Cycle to the Loading Cycle definition;
• the Loading Cycle operations receiving the least processings.

Decline in the number of component-processings. The Turntable operations are reprocessed more often than the other ones: more than six component-processings vs. two to three processings for the other functions. The decline in the number of processings per operation follows the order of cycle definition. This result can not be attributed to the definition of the machine functioning being more and more constrained as specification progresses, because - except for some very few interactions (see below, §3.3.1) - each station functions independently of the other ones. The most plausible explanation for this result seems to be a learning effect.

The few Loading Cycle component-processings. The operations on the Loading Cycle receive, proportionally, the least component-processings (an average of 2,5). This result is considered in relation with a result presented earlier: the Loading Cycle definition being interrupted less frequently than the other cycle-definitions: only once every 11 vs. once every four operation-processings. The following explanation may be proposed. The Loading Function is for the engineer by far the easiest one to define because it is a type of function he knows best through his experience which especially concerns assembling installations.

The fact that the first processing of this function leads only to postponing its definition is not contrary to this explanation. The postponement is done at a stage of the specification process when the engineer still uses the example sequences schema: his difficulty in understanding the Loading Function reference in this example is due to the representation used for this function, not to the functioning represented.

Reprocessing an operation: completing or/and modifying it

• An operation may be reprocessed to complete its definition by attributing it a value on a new dimension: 40% of operation reprocessing is of this type (between 21% - W1 - and 54% - W2);
• An operation may be reprocessed to modify its definition by changing a value on a dimension already used: 44% of operation reprocessing is of this type (between 29% - W2 - and 59% - W1).
• During a same operation-reprocessing the engineer may complete and modify the definition of the operation: 10% of operation reprocessing is of this type.
• Finally, 6% of operation-reprocessings do not contribute to its definition.

The contrasting results for the two Work Cycles with regard to reprocessing the definition of an operation, that is, for completing it (W1 < W2), or for modifying it (W1 > W2), may be interpreted as follows.
The engineer starts the definition of the Work Functions by the First phase. The Second phase is, on various points, analogous to the First phase, which constitutes a reference for it. So its definition takes advantage of the processing already done on its reference. In addition to that, defining the Second phase leads the engineer to discover errors and omissions in the First phase, and thus, to reprocessings of this function.

3.3 QUALITATIVE RESULTS: PROCESSES LEADING TO PROPOSING PLAN-DEVIATION ACTIONS

To account for plan-deviation during specifying, a model of the specification activity is necessary (see Visser, 1988b, where some first elements were presented, especially with regard to control). The observed deviations do not follow another plan, one the engineer is not conscious of, for example. They are not systematic, but depend - like the occurrence of planned actions - at each moment, on the data which the engineer has: especially the state of the sequences schema in progress, his representation of this schema and his knowledge, and the information which he has at his disposal and which he receives. This, in effect, gives the activity's organization an opportunistic character.

Control. In a model of an activity, the control is the component on which depends how the activity is organized. The control is the system component which accounts for deviating from a plan - and for resuming it. To build a model, and especially to formulate the control knowledge used by the engineer, a qualitative analysis is made of the data, especially with regard to the nature of the *transitions between the different component-processings.* On this point, the remarks of the engineer are very important, but the analysis goes of course beyond the data. Having a type of model in mind, the researcher may infer - from sequences of component-processings and their transitions, and the remarks made by the engineer - the knowledge which is used for specifying and the way it is used.

A blackboard model of specification

The observations on the engineer show the specification activity to be organized opportunistically. The presence - and even the use - of a plan does not contradict this conclusion. The plan which guides the activity is used in an opportunistic way, that is, only when no more opportune actions come up (see Hayes-Roth & Hayes-Roth, 1979).

Blackboard models have been proposed for opportunistically organized activities. Their main components and control structure will be presented without any detail (see Bisseret, Figeac-Létang, & Falzon, 1988, for an application of the model to the activity of traffic signal setting; Hayes-Roth & Hayes-Roth, 1979, for an application to errand planning; Nii, 1986, for a general presentation of the model).

Main components of the model. A blackboard model has two basic components, (a) the knowledge sources partitioning the knowledge used for solving the problem, and (b) the blackboard data structure, the database holding the problem-solving state (the problem-solver's working memory). In general, a separate control component exists in systems based on the model.

Control structure. The different actions (or problem-resolution steps) - here component-processings - are articulated according to the following iterative sequence:

i. An action modifies the state of the blackboard
ii. One or several knowledge sources propose themselves as being able to make a contribution to the resolution of the problem as it is defined by the state of the blackboard

iii. The control selects, among the proposed actions, one action to be executed, according to
  - the state of the blackboard (see i.)
  - the knowledge sources proposing themselves (see ii.)
  - the control knowledge (especially the selection criteria used, see below)

iv. Back to i.

Selection of a plan-deviation action[14] rather than a planned action

Deviations of the engineer's plan are considered to be actions which have been proposed as an alternative to the planned action (for reasons analyzed below) and which have been selected by the control, rather than the planned action.

To decide which one is going to be selected for execution, the control evaluates each proposed action on, at least, two criteria. So, when several actions are in competition - that is, when, next to a planned action, one, or various, other actions are proposed - these actions are compared by the control instance. But the result of this evaluation may lead the control also to decide to not select an isolated planned action - that is, a planned action which is the only one to be proposed. In that case, another action has to be selected (see §3.3.6).

First action selection criterion: Cognitive cost. The most important criterion for selecting an action is its relative cognitive cost. For each action proposed, this cognitive cost is determined.

For evaluating the cognitive cost of an action, the control considers - not necessarily independent - factors such as:

- **the availability of a "schema" for executing the action**. A schema is a memory data structure defining a generic concept on a number of dimensions and providing, for each of these dimensions, a certain number of values, including, generally, one default value (see Rumelhart, 1978). The execution of an action, for which such a memory representation is available, may be of relatively little cost if all relevant execution dimensions have default values.

  The engineer's plan may be considered as such a schema with default values. This is the reason why its use is so interesting from the viewpoint of cognitive cost. But deviating from this plan may sometimes be still more "interesting" (that is, from the viewpoint of cognitive cost) for various reasons which will be presented below (see §3.3.1-§3.3.5).

- **the availability of the information elements required for executing the action**. On the one hand, an action may be interesting on condition that the information elements required for executing it are available, that is, accessible without many effort. If this is not the case, the cognitive cost of accessing these information elements reflects on the cognitive cost of the action - and thus, its interest.

  On the other hand, an action may become interesting because of the relative availability of the information elements required for its execution (see §3.3.1-§3.3.4).

[14] Shorthand for "an alternative-to-the-planned action proposition leading to a plan-deviation if selected".

• **the relative difficulty of the action**. This factor is, for the moment, rather vague. Yet it is mentioned because the explanation of various observed deviations suggested it (among them, the postponement of the definition of the Loading Cycle (see §3.2.1) and, once formulated, it seems a plausible factor. It needs however to be clarified.
Two examples will be presented to illustrate it.
Example1. In general, to determine the value of a descriptor, the engineer knows which type of information elements are required and in which type of information source he may find them.
The duration of operations, for example, may be determined in various ways, from *retrieving it* from a document (such as a tool-plate) to *calculating it* from its components (motor speed and advance distance/rotation). *Retrieving a value* is the easiest way to determine it, *calculating it* is rather difficult.
A still more difficult action seems to be an action for which even the required "basic ingredients" (such as "motor speed" and "advance distance/rotation" for calculating a duration) are not given (in an information source), but have to be worked out by the engineer.

Example2. If durations are given - on a tool-plate for example - they are presented, generally, for the individual operations. However, the engineer's design colleagues made only a very global requirements analysis, and sometimes several operations which the engineer defines were assembled into one global operation by these colleagues. In that case, the engineer will have to divide the corresponding global duration over the different individual operations. Rules or heuristics have been identified for this division. They will not be presented. What matters here, is that such a division action seems to be rather difficult.

NB1. The high cognitive cost of a planned action - even an action which is not in competition with any other proposed action - may lead the control to not select it, and to skip it (see below §3.3.6).

NB2. The cognitive cost of a deviation is a combination of the cost of the corresponding plan-deviation action **and** the cost of the plan-resumption.

Second action selection criterion: Importance. The second criterion is "importance". Actions differ on this point. Their importance depends on:
• the importance of the type of action
Example. As we know the engineer's plan, actions may be identified as plan-deviation actions. So, when the engineer, discovering a hitherto forgotten operation, proceeds - in spite of his plan - to the definition of this operation, this action is identified as a plan-deviation action. The nature of the action (here "fixing an omission") and of the component concerned (here an "operation") + the possible remarks of the engineer are used to explain for the deviation. For example, one of the (control) knowledge elements which are formulated to account for this (and other) observations, is the following:

Fixing the omission of an operation which has been forgotten is an important action.

• the importance of the object concerned by the action
Example. "Verifying" is an important action if it concerns "durations", but not if it concerns "identifiers". That means that to verify the duration of an operation[15], the engineer frequently deviates from his plan, but never to verify its identifier.

Resulting selection of an action. Other criteria may intervene. They have not been identified. Neither has the way in which the two criteria are combined been identified: if an action, proposed as an alternative to the planned action, is important and/or if its cognitive cost is relatively low compared to that of the planned action, this action is selected for execution.

Distance between the component abandoned and the component the deviation leads to

In the presentation of the different processes leading to plan-deviation, three types of plan-deviation actions may be distinguished, according to the distance between the component concerned by the current processing and the

[15] To verify the duration of an operation, the engineer recalculates it from the information elements which his mechanical design colleagues used for calculating them, that is, the technical specifications of the motors or tools concerned (their rotation speed and their advance distance/rotation).

component concerned by the plan-deviation action (hereafter referred to as a "deviation-action").

Inter-component deviations. Most deviations stay within the cycle that is currently being defined, going from one of its components to another.

Two other types of deviations have been observed:

Inter-cycle deviations. Some deviations go from a component in one cycle to an - often related - component in another cycle. The cycle to which the deviation leads is, generally, an already defined cycle - even if its definition is incomplete and/or imperfect.

Intra-component deviations. Some deviations go from the descriptor of a component to another descriptor of this component. They constitute deviation-actions in so far as they go against the planned order of dimensions on which a component is defined.

* * *

After this introduction, presenting the conceptual context of plan-deviation, the different underlying processes will be described.

Theoretically, these processes which have been identified as leading to deviation-actions are only a first condition for a plan-deviation to occur. They lead to *proposing* actions. The second condition is that the action which has been proposed is *selected*. That is, these processes lead to alternative-to-the-planned action propositions. Only if the proposed action is selected will they have led, albeit indirectly, to a plan-deviation.

The processes presented are inferred mostly from observed real deviations. Processes leading to proposing actions which were not selected afterwards, or processes which could have led to proposing actions, but which did not, were simply not observed: the only possibility for identifying them - otherwise than by observing them - would have been for the engineer to have formulated a remark disclosing them. So, the processes which are presented led in fact, with some rare exceptions, to deviation-actions.

They are probably not specific for one type of deviation, that is, they may lead to inter-component deviations, to inter-cycle deviations, and to intra-component deviations. But most deviations go from a function operation to another operation of the same function - that is, are inter-component deviations -, so most processes have been identified as being this type of deviation, and will be illustrated by examples of it.

3.3.1 A component's mental representation activating another component's representation

The processing of the mental representation of a component Cx - for defining Cx - may lead to activate, in memory, the representation of another component, Cy.

This activation has several causes, among which the type of relationship between the components Cx and Cy[16] (see below). Leaving aside this

[16] "Relationships between components" is used here, as shorthand, for "Relationships between mental representations of components".

relationship between the components, the relationship between the type of processings on the components plays also a role: Cy may be activated by the processing of Cx as

- "being to defined also".
  Example. The engineer starting to define the Second phase work function on the General Cycle-plate deviates from his plan by not finishing this definition in order to define - also only partially - the First phase work function, its analogous function (see below).

- "having been defined erroneously" or "having been defined incompletely"
  Example.The engineer defining the Fast Advance Movement on the Second phase work cycle interrupts his plan in order to verify the corresponding Fast Advance Movement on the First phase work cycle (already defined, if only partially). He thinks that he has forgotten a starting condition for this First phase Movement - "starting conditions" was the dimension on which he was defining the Second phase Movement when he abandoned his plan.

If the control selects the corresponding proposition of deviation-action for defining or fixing Cy, it is because - if the corresponding processing of Cy is done *now* - advantage may be taken of the processing just done on Cx.

Four different types of relationship between components leading to this type of deviation have been identified:

Analogy (or correspondence). This type of activation has been observed in nearly all cases between Work components, either between the two Work Functions themselves or between two corresponding operations on the Work Functions.

Example. The engineer defining a Tool Compensation operation in the Second phase work cycle, returns to the just defined First phase work cycle, to complete its corresponding Tool Compensation operation. The completion consists of adding the starting conditions corresponding to those which he has just defined for the Second phase Tool Compensation operation.

> NB. The engineer deals with the First phase workstation cycle before the Second phase workstation cycle, and he often returns, for reasons of analogy, from the Second to the First phase.
> The programmer works in the opposite direction (partly because he is guided by an "example" program which gives the corresponding modules in that order). Thus the programmer writes sub-modules on the First phase workstation using as an example the corresponding Second phase sub-module, which he has already written.

Prerequisites. Several times, the engineer discovers an operation which he and/or his colleagues had forgotten - which is mostly a Prerequisite operation (see §3.2.2) -, when he is defining the Work operation of which it is a prerequisite.

Example. The definition of the Tooling-by-a-Backward Movement Work operation[17] leads the engineer to discover that he forgot two of its prerequisites: the Tool Retract Movement and the Tool Compensation.

> NB. The specific role of prerequisites in information processing was also noted in a study on functional analysis carried out by student automaticians in a programmable controller program design task. These subjects modified the analysis method they had learned (based on the Grafcet formalism) in order to adapt it to their mental representation of the functioning they had to analyze. In particular, they did this by processing

---

[17] See the § Operations forgotten are Prerequisites in section 3.2.2.

the aspects of the process directly related to its goal (≈ Work operations) before its prerequisites (≈ the Prerequisite operations of the Work operations).

The first ones - that is, actions leading directly to the goal - were considered right from the first problem-solving stage. The second ones - that is, actions preparing the process to attain its goal, the prerequisites of the first ones - were handled in a late processing stage, or even completely omitted (see Morais & Visser, 1985).

Interaction. This type of relationship may be based, as in the example above, on a relation between two physical units in the installation (for example, the turntable interacts with each of the three other stations as these must be retracted before it can turn). The engineer starts to define the Turntable Cycle - and defines it partially - before the other Function Cycles. During their definitions, he mostly goes back to the Turntable cycle in order to introduce definition elements on it which are related to the interaction between the current functioning and the turntable functioning. Generally he adds ending conditions of the operations of the Function Cycles in question to the starting conditions of the first operation of the Turntable[18] - , but he also fixes errors which he has discovered in these conditions.

Example. Defining two Pliers Advance operations on the Loading Cycle, the engineer discovers that he has forgotten to introduce the corresponding Pliers Return operation ending conditions among the starting conditions of the first operation of the Turntable, Release Grip Piece[19].

At the level of operations, another type of interaction plays a role. The starting conditions of an operation comprise always one or more of the ending conditions of its preceding operation(s). These two-directionial relationships lead to two different deviation-patterns. On the one hand, the processing of the starting conditions of an operation has been observed to lead to the discovery - and fixing - of omissions or errors on the ending conditions of its preceding operation. On the other hand, the processing of the ending conditions of an operation has lead the engineer to process, in anticipation, the following operation, in order to define its starting conditions.

Example. The definition of the starting conditions of the Turntable Fast Movement is interrupted to complete the ending conditions of the Turntable Release Grip Piece operation. Later on, after a modification of the Release Grip Piece ending conditions, the engineer reflects this modification onto the Fast Movement starting conditions.

Opposites. Dealing with an operation, the engineer sometimes discovers that he has omitted its opposite operation.
Example. Defining a Tool Return Movement, the engineer discovers that he has forgotten the corresponding Tool Advance Movement.

3.3.2 Defining a component on dimensions Di-Dn leading to a local plan for defining other components on the same dimensions

The presentation of the deviation strategies described above focused on the relation between components representations. In this section, an "analogous" deviation strategy is presented. The relationship is between the dimensions on which the components are defined. The components to which the deviation

---

[18] See notes 4 and 8.

[19] The Pliers take the pieces for transferring them to the Conveyors.
These conditions are starting conditions for the Turntable, not for its first operation, but see Note 8.

leads have no relation with the currently defined component other than belonging to the same function. The strategy consists in processing one or more components on definition dimensions which are related to the dimensions currently used - and so it leads to plan-deviation actions, and frequently to several plan-deviation actions consecutively.

Example. This strategy, which is responsible for numerous deviation-actions, is the most important source for deviation on the Loading Cycle. The engineer starts the definition of this cycle by defining its first five operations on the couple of dimensions "identifier" and "starting conditions". After some operation-processings conducted according to the plan, he defines four consecutive operations on the descriptors triplet "identifier"-"starting conditions"-"ending conditions", followed - after some planned actions - by defining four other operations on the couple "identifier"-"duration".
When he really defines the Loading Cycle - not during its first failed definition-session - this definition is the first and only one[20] for which the engineer no longer refers to the "example" sequences schema, and for which his most important information source is his knowledge. This could explain this strategy. That is, if the engineer relies mostly on his knowledge, that is on his memory, processing would proceed by way of "dimension-definitional correspondence" : having defined an operation on some dimensions, the engineer makes the corresponding definition contribution to other operations. But if this way of processing is very "spontaneous" or "natural" for the engineer, what about his plan, in which the unity of processing is not a "definition-dimensional grouping", but the component (in the present case, the operation)?

This type of plan-deviation has also been observed as a consequence - or a corollary - of taking advantage of the information elements available (see below, §3.3.4). Using an information source for defining a component on some dimensions leads the engineer then to formulate a local plan for using this information source also to define other components on the same dimensions, because it provides - for several operations - these dimensions, but no others - so it cannot be used for finishing the definition of the component currently being processed.

Example1. This data-driven definition strategy - from the information elements on the consulted information source to the components which may be defined by way of them - may have led the engineer to define, consecutively, the two Work functions of the General Cycle, first both on the dimension "duration" - taking advantage of his working memory, and then both on the dimension "identifier" - taking advantage of the tool-plates.

Example2. If, for the plan-deviation presented in Example1, the relationship between both functions may also have played a role, this possibility is absent for the following deviation.
For the first component-processings of the Turntable Cycle, the engineer uses a document representing its operations with their duration. So, the four Turntable operations are first defined, each one consecutively, on the two dimensions "identifier" and "duration", before the engineer looks elsewhere for information elements allowing them to be defined on the other definition-dimensions.

3.3.3 Processing an information element from different points of view

An information element, used for defining a component, may be processed from the point of view which allows it to define another component.

For example, an information element, used for functional definition, may be considered from its mechanical (physical) point of view.

Example1. To define the starting conditions of the first operation of the Turntable Cycle, the engineer consults a mechanical specifications document presenting, among other information elements, all the electrical detectors on the station (the activation of which may constitute operations starting - or ending conditions). Processing these information elements, the engineer comes to question the mechanical safety of the Turntable: he is not sure that the Turntable, during its rotational movement, will not hit other stations. So he interrupts his cycle definition - he even interrupts his functional specification activity - to verify the mechanical specifications.

[20] The Conveyors Cycle will also be defined - only in the Grafcet formalism - without the use of another Conveyors Cycle representation.

Example2. Looking for the ending conditions of the Loading function, the engineer discovers that not all detectors required for giving the corresponding information (that is, that the operation is finished) are specified on the mechanical specifications document. This leads him to conclude that the operation is not completely controlled on the installation such as it has been designed, and he interrupts the current definitional processing of the Loading function to complete the mechanical design of the installation.

Another example of this type of information processing is the interpretation of an information element which allows a component to be defined on one dimension, in such a way that it may also contribute to the definition of another dimension. For the dimensions used in the observed definitional activity, this occurs only for the starting and ending conditions.

Example. Retrieving an information element for defining the ending conditions of an operation On, the engineer also frequently uses it to define the starting conditions of On+1.

3.3.4 Taking advantage of information elements available

Acting according to a plan is a concept-driven activity. Starting from a goal imposed by the plan, the engineer will look for information elements allowing him to satisfy this goal. Most deviations observed however stem from a data-driven processing, such as starting from information elements which the engineer has at his disposal and which allow him to satisfy a goal that is not plan-imposed.

Information elements are generally there because they are used for the current component-definition (see above the possible deviational use in this case), but they may also be available because the relevant information source "presents itself" to the engineer. Some examples of such information presentation are the following:

- the client communicates modified requirements to the engineer
  Example. There is no example of the client changing the requirements during the construction of the sequences schema. Some have been observed during the construction of the Grafcet (see Morais, 1987).
- a colleague presents him with new information, especially in relation to the mechanical specifications
  Example. When the engineer consults a colleague about the use of three detectors - represented on a mechanical specifications document - for giving the ending conditions of an operation, he learns that one of the three is no longer necessary. So the engineer uses only the other two. This leads him to verify the operations for whose definition these elements had been used already, and to fix the errors identified.[21]
- a specialist comments on the current problem solution-state attained by the engineer
  Example. At a rather advanced stage of the functional specification, the engineer learns from a design colleague - an electrical engineer - that, if one uses a particular type of detector, one must control the presence of the operation for which they are used AND the absence of the corresponding opposite operation. This leads him to verify the already defined operations controlled by this type of detector, and to add the missing ending conditions.
- the information source the engineer consults comprises modifications which "stick out a mile"
  Example. Most information sources used by the engineer are also used by colleagues. So, modifications may have intervened from one consultation to another. The tool-plates are such a changing document which is worked on, and consulted, by various persons. In addition, the engineer consults only a very restricted area of them, but the information elements in this area are very important and the engineer knows the values of some of them by heart.
  Consulting the tool-plates for the definition of the General Cycle, the engineer remarks that the durations of the Work functions have been changed: these are "obvious" modifications for him, because they concern values which he knows inside out. Taking these modified values into account, the engineer interrupts his current processing to redefine the two Work functions.

---

[21] This is not the place to comment on the communication problems - especially the absence of communication - between the different persons involved in a same design project.

3.3.5 Drifting (during a "difficult" specification-action)

"Drifting" - that is involuntary attention switching to a processing other than the current one - was observed to occur especially when the engineer was involved in a "difficult" action, that is, in general, trying to determine the value of a component on one of its definition-dimensions (see §3.3, section "Selection of a plan-deviation action rather than a planned action").

Example. The engineer, involved in his definition of the Turntable Fast Movement - whose ending conditions he finds hard to define - is observed to interrupt and to define the ending conditions for the following operation, remarking: "There, I know what to put".
Without this remark, the observed deviation could have been explained as due to "Postponing an action which costs too much - if executed now" (a process presented below, §3.3.6).

One might think that during a "difficult" definition action, all attention is focused on the current problem. Perhaps drifting, in these conditions, may be explained by the possibility that the engineer - looking in various directions for possible solution elements - comes upon an information element whose application to another component definition is "obvious" (see above, §3.3.4). A hypothesis inspired by the observations is that these drifting-caused deviations occur especially during information retrieval for problem-solving that is not guided by strict information searching rules.

Example. In the example presented above, the engineer was observed to "think and hesitate": "What shall I put there? ... I don't know", rather than to wonder which information elements were required for defining the ending conditions and which information source(s) should be consulted.

3.3.6 Postponing actions which cost too much - if executed now

The cognitive cost of each action is evaluated before its - possible - execution. Frequently, an important alternative action is "cheaper" than a planned action, and therefore selected. This is how plan-deviation generally occurs. But sometimes, a planned action is skipped because it costs too much, not compared to another action, but to the action itself when executed afterwards. In addition to selecting another action, a local plan is then formed for reproposing the postponed action as soon as the conditions leading to its postponement are no longer satisfied.
Example. The definition of the Loading cycle was skipped and postponed, because the engineer did not understand the decomposition it had received in the example he used for the specification of the installation (see §3.2.1).

A planned action may also cost too much because the required information elements are unavailable (see §3.3, section "Selection of a plan-deviation action rather than a planned action"). This may be the case, for example, because

- the client has not yet taken a decision required for defining the component in question (and the engineer judges that this decision may not be taken by him)
  Example. This is the reason why the engineer did not define the Conveyor cycles until his construction of the Grafcet.
- the colleague who has the relevant information is absent
  Example. The definition of the ending conditions of an operation which is controlled by a hydraulic control detector is skipped because the hydraulic specialist is absent.

* * *

This last section - presenting processes leading to plan-deviation - focused on the control component in a blackboard system. The structure of the blackboard, and the knowledge sources have not been presented, other than allusively. To study these points, the structure of the engineer's knowledge representation should be examined. Some possible directions have already been sketched, through the observation that deviations may go from component to component, for example, by way of relationships

- between (the referents of) their respective representations (analogy, prerequisites, etc.)
- between the type of definitional processing they may undergo (definition on the dimensions Di-Dn).

Such results suggest representation units with mutual links along different types of dimensions, but this point needs to be examined more precisely.

4. CONCLUSION

The results which have been presented show the specification activity to be an opportunistically organized design activity. It is a typical design activity, because

- the problem to be solved is not defined completely, and immutably, right from the start: an important part of the engineer's activity consists of constraining this problem;
- the solution to be obtained is not unique: the "final" specifications arrived at by the engineer may be - and are going to be[22] - modified, not because they are "incorrect", but because an alternative design may be chosen by another designer, using other criteria;
- the activity does not follow a pre-existing plan.

The opportunistic organization of the activity was the result on which this paper focused. The engineer had a - hierarchically structured - plan for his activity, but he used it in an opportunistic way: he used it as long as it was interesting from the viewpoint of cognitive cost, that is, if more cognitive economical actions came up, he abandoned it.

Due to the control it allows, a blackboard model is interesting for modeling such an activity. In this context, deviations from the engineer's plan were considered to be actions which have been proposed as an alternative to the planned action and which have been selected by the control instance, rather than the planned action.

Processes leading to alternative-to-the-plan action propositions have been presented, together with the most important criteria used by the control for selecting the corresponding actions.

Implications for design assistance tools (CAD). If the design activity is opportunistically organized, a system which supposes a hierarchically structured design process will at least constrain, but probably even handicap the designer - and thus not assist him at all (see Visser & Hoc, in press). Hoc (1988a), in an evaluation study of a programming environment supporting top-down processing, showed that professional programmers - trained in the underlying structured, top-down programming method - experienced difficulties due to the processing imposed by the environment, and generated non-optimal solutions.

The results which have been presented here constitute a strong argument for tools allowing the problem-solution to be abandoned at a certain level, in order to process solution elements at another level, and to possibly resume afterwards

---

22 The person who is in charge of debugging and testing the program for the control part modifies the specifications of the manual operation mode in order to give the installation "a more flexible - manual - use".

the solution state at the level abandoned. Such tools should not, however, *impose* such a resumption. The engineer observed has got - and uses - a hierarchically structured plan, to which he returns after more or less deviation-actions. Not all designers do necessarily refer to such a plan - nor do all design activities, especially if their result has a less constrained structure than a sequences schema.

5. REFERENCES

Anderson, J. R. (1986). Knowledge compilation: the general learning mechanism. In R. S. Michalski, J. G. Carbonell & T. M. Mitchell (Eds.), Machine learning. An artificial intelligence approach (Vol. II). Los Altos, Calif.: Morgan Kaufmann Publishers.

Bisseret, A., Figeac-Létang, C., & Falzon, P. Modeling opportunistic reasonings: the cognitive activity of traffic signal setting technicians (Research Report N° 898). Rocquencourt: INRIA, 1988.

Ericsson, K. A., & Simon, H. A. Protocol analysis. Verbal reports as data. Cambridge, Mass.: MIT Press, 1984.

Hayes-Roth, B. & Hayes-Roth, F. A cognitive model of planning. Cognitive Science, 1979, 3, 275-310.

Hoc, J. M. Towards effective computer aids to planning in computer programming. Theoretical concern and empirical evidence drawn from assessment of a prototype. In G. C. van der Veer, T. R. G. Green, J. M. Hoc, & D. Murray (Eds.), Working with computers: theory versus outcomes. London: Academic Press, 1988a.

Hoc, J.M. Cognitive psychology of planning. London: Academic Press, 1988b.

Morais, A. Hierarchical planification in programming. In I. D. Brown, R. Goldsmith, K. Coombes, & M. A. Sinclair (Eds.), Ergonomics International 85. Proceedings of the Ninth Congress of the International Ergonomics Association, Bournemouth, England, 2-6 September 1985. London: Taylor & Francis, 1985.

Letovsky, S., Pinto, J., Lampert, R., & Soloway, E. A cognitive analysis of a code inspection. In G. Olson, S. Sheppard & E. Soloway (Eds.), Empirical Studies of Programmers: Second Workshop. Norwood, N.J.: Ablex, 1987.

Morais, A. Acquisition d'un outil de spécification (GRAFCET) pour décrire un procédé automatisé (Rapport de Recherche INRIA n°631). Rocquencourt: INRIA, 1987.

Morais, A., & Visser, W. Etude exploratoire de la programmation d'automates industriels chez des élèves de l'enseignement technique (Rapport de Recherche INRIA n°404). Rocquencourt: INRIA, 1985.

Newell, A., & Simon, H. A. Human problem solving. Englewood Cliffs, N.J.: Prentice-Hall, 1972.

Nii, H. P. Blackboard systems: the blackboard model of problem solving and the evolution of blackboard architectures. The AI Magazine, 1986, Summer and August.

Rumelhart, D. Schemata: the building blocks of cognition. In R. Spiro, B. Bruce, & W. Brewer (Eds.), Theoretical issues in reading comprehension. Hillsdale, N.J.: Erlbaum, 1978.

Sacerdoti, E. D. Planning in a hierarchy of abstraction spaces. Artificial Intelligence, 1974, 5, 115-135.

Visser, W. Strategies in programming programmable controllers: a field study on a professional programmer. In G. Olson, S. Sheppard & E. Soloway (Eds.), Empirical Studies of Programmers: Second Workshop. Norwood, N.J.: Ablex, 1987.

Visser, W. Giving up a hierarchical plan in a design activity (Research Report N° 814). Rocquencourt: INRIA, 1988a.

Visser, W. Towards modelling the activity of design: an observational study on a specification stage. Preprints for Proceedings of the IFAC/IFIP/IEA/IFORS Conference Man-Machine Systems, Analysis, Design and Evaluation (Oulu, Finland, 14-16 June 1988) (Volume 1). IFAC, 1988b.

Visser, W., & Hoc, J.M. Program design strategies. In D. Gilmore, T. Green, J.M. Hoc, & R. Samurçay (Eds.) Psychology of programming. London: Academic Press, in press.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique